DELIFT: DATA EFFICIENT LANGUAGE MODEL INSTRUCTION FINE-TUNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Fine-tuning large language models (LLMs) is essential for enhancing their performance on specific tasks but is often resource-intensive due to redundant or uninformative data. To address this inefficiency, we introduce DELIFT (Data Efficient Language model Instruction Fine-Tuning), a novel algorithm that systematically optimizes data selection across the three key stages of fine-tuning: (1) instruction tuning, (2) task-specific fine-tuning (e.g., reasoning, question-answering), and (3) continual fine-tuning (e.g., incorporating new data versions). Unlike existing methods that focus on single-stage optimization or rely on computationally intensive gradient calculations, DELIFT operates efficiently across all stages. Central to our approach is a pairwise utility metric that quantifies how beneficial a data sample is for improving the model's responses to other samples, effectively measuring the informational value relative to the model's current capabilities. By leveraging different submodular functions applied to this metric, DELIFT selects diverse and optimal subsets that are useful across all stages of fine-tuning. Experiments across various tasks and model scales demonstrate that DELIFT can reduce the fine-tuning data size by up to 70% without compromising performance, offering significant computational savings and outperforming existing methods in both efficiency and efficacy.

1 INTRODUCTION

Fine-tuning large language models (LLMs) is pivotal for adapting these powerful architectures (Devlin et al., 2019; Brown et al., 2020a; Touvron et al., 2023) to specialized tasks such as intricate reasoning, precise question-answering, and the seamless integration of new information (Ouyang et al., 2022). This transformation—from a general-purpose model to a task-specific agent—heavily relies on the quality and nature of the data employed during fine-tuning, which critically determines the model's subsequent performance (Wei et al., 2022; Zhou et al., 2023; Hoffmann et al., 2024).

The effectiveness of fine-tuning hinges on the **quality**, **diversity**, and **relevance** of the selected data (Gururangan et al., 2020; Wei et al., 2022; Zhou et al., 2023). High-quality data ensures accurate learning, diverse data enhances generalization, and relevant data aligns the model's capabilities with specific application needs. However, optimizing data selection across different fine-tuning phases remains a significant challenge, leading to our central research question:

How can we create a unified framework for efficient data selection across all fine-tuning stages of LLMs, while optimizing performance and maximizing data efficiency?

To address this challenge, we present DELIFT (**D**ata **E**fficient **L**anguage model **I**nstruction **F**ine-**T**uning), a novel, unified, and computationally efficient algorithm engineered to optimize data selection across all stages of the fine-tuning process. The key innovation of DELIFT lies in its pairwise utility metric, which assesses the informational value of data samples relative to both the model's current capabilities and other samples within the dataset. This metric, combined with submodular optimization techniques, allows DELIFT to efficiently select optimal data subsets that precisely address the model's learning requirements without incurring unnecessary computational costs.

The typical fine-tuning process comprises three key stages: **1. Instruction Tuning**: Enhances the model's ability to follow general instructions (Mishra et al., 2022; Wei et al., 2022; Longpre et al., 2023); **2. Task-Specific Fine-Tuning**: Refines the model's expertise in specific domains (Gururangan et al., 2020; Cobbe et al., 2021); **3. Continual Fine-tuning**: Enables the model to integrate new information while mitigating catastrophic forgetting (Madotto et al., 2021; Wu et al., 2024). DELIFT is able to optimize data selection processes across all three stages. Additionally, DELIFT offers significant benefits for In-Context Learning (ICL) (Brown et al., 2020b; Xue et al., 2024). By utilizing the selected subsets as the ICL example pool, DELIFT achieves similar or better performance compared to using the entire dataset, thereby enhancing data efficiency in ICL scenarios. This dual functionality is empirically validated in our experimental results.

Existing data selection methodologies often fail to address the nuanced requirements of the aforementioned distinct fine-tuning stages. Many approaches are tailored to a single stage, lacking the adaptability needed for comprehensive fine-tuning (Xia et al., 2024; Liu et al., 2024; Bukharin & Zhao, 2024; Chen et al., 2024). Others depend on computationally intensive procedures, such as exhaustive gradient computations, rendering them impractical for large-scale models and datasets (Killamsetty et al., 2021b;a; Xia et al., 2024; Zhang et al., 2024). Additionally, some methods utilize features obtained from an independent model that are not specifically aligned with the model undergoing fine-tuning, reducing their effectiveness (Killamsetty et al., 2023; Liu et al., 2024; Bukharin & Zhao, 2024; Chen et al., 2024; Du et al., 2023).

DELIFT addresses these limitations by adapting to the unique requirements of each fine-tuning stage. **1. Instruction Tuning**: Selects diverse data to enhance general instruction-following capabilities; **2. Task-Specific Fine-Tuning**: Prioritizes data that is aligned with the target task, to refine specialized expertise; **3. Continual Fine-tuning**: Identifies novel, complementary information to expand the model's knowledge base while safeguarding against catastrophic forgetting.

Figure 1 illustrates how DELIFT optimizes data selection across these stages, demonstrating the selection and pruning processes in each fine-tuning phase. By leveraging submodular optimization techniques (Fujishige, 2005; Bilmes, 2022) and submodular information measures (Iyer et al., 2021), DELIFT efficiently selects optimal data subsets that precisely address the model's learning requirements without incurring unnecessary computational costs. This approach effectively balances data utility and computational efficiency.

Our key contributions are as follows:

1) Versatile Pairwise Utility Metric: A novel, easy-to-compute metric for assessing data informativeness, incorporating model feedback applicable across all fine-tuning stages.
2) Unified Data Selection Algorithm: DELIFT systematically optimizes data selection for instruction tuning, task-specific fine-tuning, and continual fine-tuning within a single framework.
3) Computational Efficiency: Circumvents resource-intensive operations, ensuring scalability to large datasets and models. DELIFT achieves at least 70% reduction in computational time compared to gradient-based methods on benchmark tasks.
4) Enhanced Performance with Reduced Data: Demonstrates the ability to reduce fine-tuning data size by up to 70% without compromising performance, and achieves comparable efficacy as to utilizing the full dataset.
5) Improvement over Existing Methods: Outperforms current data selection techniques by up to 26% in effectiveness across diverse tasks and model scales (see Section 4).

The remainder of this paper is organized as follows: Section 2 provides background on fine-tuning LLMs and reviews related work. Section 3 details the methodology behind DELIFT, including the development of our pairwise utility metric and the submodular optimization process. Section 4 presents experimental results that showcase the effectiveness and efficiency of our method. Section 5 discusses the implications of our findings and potential future directions. Finally, we release our code base for further research.

2 Related Work

Efficient data subset selection is vital for enhancing training efficiency in deep neural networks while maintaining or improving model performance. This section categorizes existing subset selection methods into model-independent and model-dependent approaches and identifies the gaps our work addresses. ***Model-independent subset selection*** methods focus on selecting representative subsets without model-specific feedback. Common approaches include using pre-trained sentence

Selected in the subset

Pruned out of the subset

Use Case 1: fine-tune a model to follow instructions. Subset should contain points that are **diverse**.

Dataset

Instruction	Input	Output
Given the context, answer the question.	**Question**: Who is New Zealand's Prime Minister? **Context**: Christopher Mark Luxon has served as the 42nd prime minister of New Zealand since November 2023.	Christopher Mark Luxon
Given the context, answer the question.	**Question**: When did Luxon start his term? **Context**: Christopher Mark Luxon has served as the 42nd prime minister of New Zealand since November 2023.	November 2023
Write a sentence with the given words.	Sun, park, dog.	Once the sun was up, I went to the park with my dog.
Classify the given objects into a category.	Crab, tuna, lobster.	Seafood.

(a)

Use case 2: improve model's performance on a mathematical reasoning benchmark. Subset should contain points that are **diverse *and* representative of the benchmark.**

(Example) Benchmark Data

Input	Output
Abby worked for 8 hours per day for 30 days. How much did she work?	240 hours
Ben paid for his dinner ($20), Charles' dinner ($18) and Dennis' dinner ($15). How much did he pay?	$53
Eunice has 20 oranges, and 4 friends. How many oranges does each friend get?	5 oranges
Greg has 20 baseball cards and trades 5 of them. How many are left?	15 cards

Dataset

Input	Output
Hannah had 40 nickels and won 10 more. How many nickels does she have?	50 nickels
Fred had 25 roses and gave 10 to Mom. How many are left?	15 roses
Lydia gave away ½ her pie to Mike and ¼ of her pie to Ned. How much of the pie is left?	¼ of the pie
Is the following word positive or negative? "Happiness"	Positive

(b)

Use case 3: continual learning on review sentiment analysis datasets. Subset should contain points that are **diverse *and* complementary to Phase I data.**

Previously Trained, Phase I Data

Input	Output
This restaurant has good paella except that it is sometimes too salty.	Negative
The waiters are impatient and rude, they rushed me to order my food.	Negative
The atmosphere of this restaurant is cozy and comfortable, with dim lights.	Positive
The food came very quickly.	Positive

New, Phase II Data

Input	Output
The fried rice is amazing!	Positive
The camera resolution quality is low, and the lens do not focus properly.	Negative
This phone is lightweight, thin, and fits in my pockets easily.	Positive
The restaurant closes too early.	Negative

(c)

Figure 1: DELIFT data selection across fine-tuning stages. (a) **Instruction Tuning**: Diverse instructions selected; redundant samples pruned. (b) **Task-Specific Fine-Tuning**: Mutually informative (with benchmark data) and diverse samples are prioritized for selection. (c) **Continual Fine-tuning**: New samples that are novel are integrated; new samples with overlapping information are pruned.

embeddings with distance or clustering metrics (Bukharin & Zhao, 2024; Sorscher et al., 2023; Killamsetty et al., 2023; Du et al., 2023; Bhatt et al., 2024), as well as employing large models like GPT-4 or pre-trained reward models for high-quality data filtering (Du et al., 2023; Chen et al., 2024). However, these methods often struggle to translate the assessed diversity or quality into downstream utility. ***Model-dependent subset selection*** aims to identify data samples beneficial to the downstream model by analyzing features like per-sample gradients or loss values. Methods such as GradMatch (Killamsetty et al., 2021a), CRAIG (Mirzasoleiman et al., 2020), and TAG-

COS (Zhang et al., 2024) focus on selecting samples that approximate the gradient updates over the full dataset. GLISTER (Killamsetty et al., 2021b) employs bilevel optimization to align gradients from selected subsets with those of a validation set. LESS (Xia et al., 2024) proposes computing gradients through LoRA fine-tuning to reduce the computational cost of gradient computation and utilizes random projection to address gradient dimensionality issues. Li et al. (2024) proposed the IFD score, a computationally efficient model-dependent metric that assesses instruction difficulty to filter challenging samples, though it does not guarantee data diversity. While effective in capturing useful samples, these methods often face computational challenges, especially with LLMs. Persistent limitations across these methods include: (i) **Limited Adaptability** across different fine-tuning stages, (ii) **Computational Intensity** due to model feedback reliance, (iii) **Lack of Unified Solutions** applicable across all fine-tuning phases, and (iv) ineffective **Redundancy Handling**. DELIFT addresses these limitations through a novel pairwise utility metric, which effectively aligns with the model's evolving capabilities throughout fine-tuning. By integrating submodular optimization with pairwise model-dependent metrics that evaluate relative sample utility, DELIFT minimizes redundancy while maximizing adaptability and computational efficiency. This approach proves effective across diverse use cases including instruction tuning, task-specific fine-tuning, continual fine-tuning, and In-Context Learning (ICL), offering a versatile and scalable solution for data subset selection.

3 METHODOLOGY

This section presents foundational concepts and the specific approach of DELIFT, focusing on data subset selection through a utility-based kernel integrated with submodular optimization techniques.

3.1 NOTATION

Let $\mathcal{D}$ denote the fine-tuning dataset, comprising elements $d_i = (x_i, y_i)$, where x_i is the input sequence and y_i is the corresponding output sequence. Our objective is to select a subset $\mathcal{A} \subseteq \mathcal{D}$ that maximizes the model's performance while minimizing computational costs. The selection strategy adapts based on the fine-tuning objective, which may include instruction tuning, task-specific adaptation, or continual learning.

3.2 UTILITY-BASED KERNEL

At the core of DELIFT lies the **utility-based kernel**, a mechanism designed to quantify the informativeness of one data point when used as an in-context example for another. As formalized in Theorem 1, this kernel has deep connections to information theory through conditional pointwise mutual information (PMI). Consider two data points, (x_i, y_i) and (x_j, y_j). The utility of data point j relative to data point i, denoted as UF_{ij}, is defined as:

$$UF_{ij} = d(GT_i, p(y_i \mid x_i)) - d(GT_i, p(y_i \mid x_i, x_j, y_j)), \tag{1}$$

where $d(\cdot, \cdot)$ is a distance metric between two probability distributions; GT_i is the ground truth distribution for the sequence y_i, assigning probability 1 to y_i and 0 to all other sequences; $p(y_i \mid x_i)$ is the model's predicted probability for y_i given only the input x_i; and $p(y_i \mid x_i, x_j, y_j)$ is the predicted probability for y_i when the model is provided with (x_j, y_j) as an in-context example along with x_i.

As shown in Theorem 1, when using KL-divergence as the distance metric, the utility metric UF_{ij} is equal to the conditional PMI between y_i and (x_j, y_j) given x_i:

$$UF_{ij} = \mathrm{PMI}(y_i;\, x_j, y_j \mid x_i) = \log \frac{p(y_i \mid x_i, x_j, y_j)}{p(y_i \mid x_i)}.$$

For practical implementation, we employ the length-normalized L2 norm (Euclidean distance) as our distance metric $d(p, q)$, given by:

$$d(p, q) = \sqrt{\frac{\sum_{k=1}^{N}(p_k - q_k)^2}{N}}, \tag{2}$$

where p_k and q_k are the k-th elements of the flattened probability distributions p and q, respectively. The normalization factor N, corresponding to the number of tokens in y_i, ensures scale invariance across different sequence lengths. We use Euclidean distance instead of KL-divergence due to practical considerations: Euclidean distance is always positive and finite, computationally efficient, and numerically stable, even when dealing with zero probabilities. This makes it suitable for subset selection algorithms like the facility location function that require positive, finite distance measures.

To compute the probability distributions accurately, we employ the **teacher forcing** technique (Williams & Zipser, 1989). This method ensures that the model uses the ground truth previous tokens when predicting each subsequent token in the sequence, enabling reliable measurement of prediction accuracy. The utility value UF_{ij} measures the improvement in prediction accuracy for (x_i, y_i) when utilizing (x_j, y_j) as an in-context example. As shown in Theorem 1, this improvement can be interpreted as approximating the sum of conditional PMI between each token in y_i and the example (x_j, y_j), given the input x_i. A positive UF_{ij} indicates that including data point j enhances the model's prediction accuracy for i by providing relevant information, whereas a negative value suggests that the example introduces misleading or irrelevant information that degrades prediction quality.

3.3 SUBMODULAR FUNCTIONS FOR DATASET SELECTION

To optimize the selection of informative data subsets, DELIFT leverages **submodular functions** (Fujishige, 2005). Submodular functions are characterized by the property of diminishing marginal returns, making them ideal for selecting diverse, informative, and non-redundant subsets. Submodular function maximization can be efficiently approximated using a greedy algorithm, with a provable approximation guarantee of $1 - \frac{1}{e}$ of the optimal solution (Nemhauser et al., 1978).

We employ three tailored submodular functions (Iyer et al., 2021), each suited to a specific fine-tuning stage:

3.3.1 FACILITY LOCATION (FL)

From an information perspective, the Facility Location function maximizes the coverage of the information space. It ensures that the selected subset $\mathcal{A}$ contains examples that are collectively representative of the entire dataset's $\mathcal{D}$ information content. This is particularly useful in instruction tuning, where we aim to capture a diverse range of instruction types and their informational content. It is defined as the following where s_{ij} is the similarity measure between data points i and j:

$$f_{FL}(\mathcal{A}) = \sum_{i \in \mathcal{D}} \max_{j \in \mathcal{A}} s_{ij}, \tag{3}$$

3.3.2 FACILITY LOCATION MUTUAL INFORMATION (FLMI)

The FLMI function is designed to maximize the mutual information between the selected subset $\mathcal{A}$ and the target domain dataset $\mathcal{D}_{\mathcal{T}}$. In our context, it ensures that the selected data points are not only informative in general but also particularly relevant to the specific task at hand. This makes it ideal for task-specific fine-tuning, where we want to bridge the gap between general knowledge and task-specific information. It is defined below where η is a scaling factor (set to 1 in our experiments):

$$f_{FLMI}(\mathcal{A}; \mathcal{D}_T) = \sum_{i \in \mathcal{D}} \max_{j \in \mathcal{A}} s_{ij} + \eta \sum_{j \in \mathcal{A}} \max_{i \in \mathcal{D}_T} s_{ij}, \tag{4}$$

3.3.3 FACILITY LOCATION CONDITIONAL GAIN (FLCG)

From an information-theoretic standpoint, the FLCG function aims to maximize the conditional information gain of the selected subset $\mathcal{A}$ given the existing dataset $\mathcal{D}_{\mathcal{E}}$. It quantifies how much new information each data point brings, conditional on what the model already knows. This is crucial for continual fine-tuning, where we want to avoid redundancy and focus on novel, complementary information that expands the model's knowledge base without unnecessary repetition. It is defined as the following where ν is a scaling factor (set to 1 in our experiments).

$$f_{FLCG}(\mathcal{A} \mid \mathcal{D}_E) = \sum_{i \in \mathcal{D}} \max\left(\max_{j \in \mathcal{A}} s_{ij} - \nu \max_{k \in \mathcal{D}_E} s_{ik}, 0\right), \tag{5}$$

Each submodular function, when combined with our utility-based kernel, guides the selection of data subsets tailored to the specific fine-tuning stage. This ensures that DELIFT selects the most informative and diverse examples, maximizing the efficiency and effectiveness of fine-tuning.

3.4 Utility Kernel as Feature Space

Our approach utilizes the **utility-based kernel** as a feature space for data selection, representing a significant departure from traditional semantic similarity-based methods. Traditional methods often rely on sentence embeddings (SE) to capture static semantic similarities between data points. In contrast, our utility-based kernel measures the actual impact of examples on model performance, providing a dynamic and task-specific assessment.

This distinction is crucial for two main reasons: **1. Semantic Diversity vs. Performance Enhancement:** While SE-based methods select diverse examples solely based on semantic content, our utility-based approach selects examples that demonstrably improve model performance across various inputs; **2. Model-Aware Selection:** The utility-based kernel is attuned to the model's current capabilities and weaknesses, enabling the selection of data points that are most beneficial for enhancing performance on the target task. By integrating the utility-based kernel with the aforementioned submodular functions DELIFT tailors the data selection process to each fine-tuning stage: instruction tuning, task-specific fine-tuning, and continual learning.

3.5 Data Subset Selection Algorithm

To operationalize our data selection approach, we employ a **greedy algorithm** that iteratively builds the subset $\mathcal{A}$ by selecting the data point that offers the maximum marginal gain in the chosen submodular function.

Algorithm 1 Greedy Maximization for Submodular Function

Require: Dataset $\mathcal{D}$, submodular function f, budget k
1: Initialize subset $\mathcal{A} \leftarrow \emptyset$
2: **for** $t = 1$ to k **do**
3: Select $d^* = \arg\max_{d \in \mathcal{D} \setminus \mathcal{A}} \left(f(\mathcal{A} \cup \{d\}) - f(\mathcal{A})\right)$
4: Update $\mathcal{A} \leftarrow \mathcal{A} \cup \{d^*\}$
5: **end for**
6: **return** $\mathcal{A}$

This greedy algorithm ensures that each addition to the subset $\mathcal{A}$ maximizes the marginal gain in the submodular function f. By iteratively selecting the most beneficial data points according to the utility-based kernel and the specific submodular function tailored to the fine-tuning stage, DELIFT efficiently utilizes the data budget to select the most informative examples.

The complete subset selection process involves the following steps: **1. Compute the Utility-Based Kernel:** Calculate UF_{ij} for all relevant pairs of data points in the dataset to assess their informativeness; **2. Select the Appropriate Submodular Function:** Depending on the fine-tuning stage (instruction tuning, task-specific fine-tuning, or continual fine-tuning), choose the corresponding submodular function (FL, FLMI, or FLCG); **3. Apply the Greedy Maximization Algorithm:** Use Algorithm 1 to iteratively build the subset $\mathcal{A}$ by selecting data points that offer the highest marginal gain according to the selected submodular function.

By synergizing our novel utility-based kernel with submodular optimization, DELIFT achieves data-efficient fine-tuning that effectively addresses both redundancy and informativeness in the data selection process, optimizing the model's performance across various tasks and domains.

Model	Qwen2						Phi-3					
Method	ICL			QLoRA			ICL			QLoRA		
	ROUGE	BGE	LAJ	ROUGE	BGE	LAJ	ROUGE	BGE	LAJ	ROUGE	BGE	LAJ
Initial	37.87	78.92	2.98	36.36	82.55	3.02	25.76	43.34	1.42	35.50	80.46	2.58
Random	39.00	80.66	3.12	44.45	85.46	3.12	33.05	72.73	2.92	44.70	83.75	2.95
SelectIT	43.08	84.50	3.18	45.14	85.88	3.21	36.11	76.31	3.18	49.68	85.84	3.20
LESS	42.08	83.24	3.26	45.16	84.95	3.28	47.10	**85.94**	3.23	48.68	**85.86**	3.24
DELIFT (SE)	47.43	84.40	3.28	48.22	86.50	3.28	46.62	85.28	3.24	45.64	83.70	3.27
DELIFT	**48.46**	**85.77**	**3.35**	**52.79**	**88.04**	**3.37**	**49.83**	85.27	**3.32**	**50.31**	84.40	**3.33**
Full Data	*58.65*	*88.72*	*3.45*	*65.51*	*92.24*	*3.51*	*55.92*	*88.26*	*3.45*	*74.98*	*93.33*	*3.84*

Table 1: Results on Use Case 1: MixInstruct. **Bold** indicates the best performance. There is a 10.44% performance percentage drop from Full Data to DELIFT after pruning 70% of the data, and a 2.27% performance percentage drop from DELIFT to the next best baseline.

Model	Qwen2						Phi-3					
Method	ICL			QLoRA			ICL			QLoRA		
	ROUGE	BGE	LAJ	ROUGE	BGE	LAJ	ROUGE	BGE	LAJ	ROUGE	BGE	LAJ
Initial	18.03	59.13	1.54	20.15	58.38	1.78	20.10	48.66	1.36	20.64	49.17	1.39
Random	20.05	59.39	1.79	20.29	59.39	1.83	20.83	49.92	2.24	24.51	53.41	2.36
SelectIT	31.38	71.08	2.86	32.96	74.76	2.90	35.37	66.67	2.52	38.98	69.84	2.54
LESS	34.59	83.23	3.07	35.03	83.37	3.50	39.69	72.12	3.17	40.32	70.89	3.24
DELIFT (SE)	34.69	83.31	3.43	35.46	83.43	3.53	37.07	71.49	3.52	38.13	79.68	3.74
DELIFT	**35.48**	**83.69**	**3.58**	**35.60**	**83.64**	**3.54**	**40.66**	**84.00**	**3.68**	**41.91**	**84.53**	**3.76**
Full Data	*36.43*	*84.25*	*3.53*	*35.88*	*76.87*	*3.63*	*42.07*	*85.26*	*3.78*	*44.73*	*87.03*	*3.82*

Table 2: Results on Use Case 1: P3. **Bold** indicates the best performance. There is *only* a 0.76% performance percentage drop from Full Data to DELIFT after pruning 70% of the data, and a 3.23% performance percentage drop from DELIFT to the next best baseline.

4 Experimental Results

We conducted extensive experiments to evaluate the effectiveness of DELIFT across various fine-tuning scenarios, model scales, and datasets. This section details our experimental setup, baselines, evaluation metrics, and results analysis.

4.1 Datasets and Use Cases

We evaluated DELIFT across the three previously described fine-tuning scenarios:

Use Case 1: Instruction Tuning We evaluated the effectiveness of DELIFT for use case 1 on two datasets: MixInstruct (Jiang et al., 2023) and P3 (Public Pool of Prompts) (Sanh et al., 2021). We randomly selected 21,000 train, 6,000 valid, and 3,000 test samples. Using the Facility Location (FL) submodular function, we aimed to select a subset of training data that was both representative and informative.

Use Case 2: Task-Specific Fine-Tuning We evaluated DELIFT for task-specific fine-tuning using two dataset pairs: (1) HotpotQA (Yang et al., 2018) with MMLU (Hendrycks et al., 2021), and (2) MixInstruct with MT-Bench (Zheng et al., 2023). We used the Facility Location Mutual Information (FLMI) submodular function to select the most informative samples from the training datasets (HotpotQA and MixInstruct) that shared relevant information with the target datasets (MMLU and MT-Bench, respectively).

Use Case 3: Continual Fine-Tuning We evaluated DELIFT in a continual fine-tuning setting using two dataset pairs: (1) SQuAD (Rajpurkar et al., 2016) paired with HotpotQA for general question-answering, and (2) proprietary query rewriting datasets covering IBM and government domains.[1] Our goal was to integrate new knowledge efficiently while minimizing redundancy. We employed

[1]In this task, non-standalone questions –questions that require previous context to answer– must be rewritten to be standalone. For example, "How much is it?" should be rewritten to "How much is the subscription for IBM Cloud?" Such queries are common in user-agent conversations where a user asks a follow-up to an agent.

Model	Qwen2	Phi-3
Method	QLoRA	QLoRA
Initial	82.10	69.10
Random	79.31	65.16
SelectIT	79.13	65.24
LESS	80.35	66.72
DELIFT (SE)	80.10	66.36
DELIFT	**81.70**	**68.70**
Full Data	*78.36*	*64.50*

Table 3: Results on Use Case 2: HotpotQA and MMLU (5-shot) for Qwen2 and Phi-3 models (classification accuracy). **Bold** indicates the best performance. For Qwen2, DELIFT outperforms Full Data by 3.34%, while for Phi-3, it improves by 4.20%.

Model	Qwen2						Phi-3					
Method	ICL			QLoRA			ICL			QLoRA		
	ROUGE	BGE	LAJ	ROUGE	BGE	LAJ	ROUGE	BGE	LAJ	ROUGE	BGE	LAJ
Initial	44.32	74.86	2.48	47.65	77.92	2.72	39.57	69.43	2.31	42.89	72.76	2.53
Random	49.78	79.54	2.83	52.91	82.67	3.05	44.63	74.28	2.62	47.85	77.39	2.84
SelectIT	54.92	83.71	3.12	57.86	86.59	3.31	49.75	78.64	2.91	52.68	81.52	3.13
LESS	59.63	85.89	3.29	62.74	88.72	3.48	54.82	81.95	3.08	57.73	84.67	3.29
DELIFT (SE)	62.85	86.94	3.38	65.83	89.76	3.57	57.69	82.87	3.17	60.54	85.59	3.38
DELIFT	**64.73**	**87.82**	**3.47**	**67.91**	**90.64**	**3.66**	**59.58**	**83.76**	**3.26**	**62.47**	**86.48**	**3.47**
Full Data	*65.89*	*88.65*	*3.55*	*69.72*	*91.53*	*3.74*	*60.76*	*84.59*	*3.34*	*64.31*	*87.42*	*3.55*

Table 4: Results on Use Case 2: MixInstruct and MT-Bench. **Bold** indicates the best performance. There is a 2.91% performance percentage drop from Full Data to DELIFT after pruning 70% of the data, and a 1.14% performance percentage drop from DELIFT to the next best baseline.

the Facility Location Conditional Gain (FLCG) submodular function, selecting complementary samples from the new dataset (HotpotQA and Government query rewrite) that provided additional, non-overlapping information to the existing dataset (SQuAD and IBM query rewrite).

4.2 Experimental Setup

Models: We evaluated DELIFT on two state-of-the-art open-source models: **Phi-3-mini-128k-instruct** (Abdin et al., 2024): 3.8B parameters, **Qwen2-72B-Instruct** (Yang et al., 2024): 72B parameters. These models were chosen to demonstrate effectiveness across different model scales.

Metrics: We use a variety of metrics to characterize performance. For n-gram word overlap we use ROUGE (Lin, 2004). For semantic similarity we calculate the dot product between the embeddings from the `bge-large-en-v1.5` model (Xiao et al., 2023); the embeddings are normalized to unit vectors, hence the closer the dot product is to 1, the more semantically similar the vectors (the metric is referred to as 'BGE'). Additionally, we use Prometheus (Kim et al., 2023), specifically the `prometheus-7b-v2.0` model, as an LLM-as-a-Judge (referred to as 'LAJ'). With our custom rubric outlined in Appendix C, Prometheus assigns scores in a range of 1 to 5 (higher scores indicate better performance.) Finally, we use classification accuracy to evaluate MMLU.

Baselines: We evaluated DELIFT by comparing it against several baselines to understand its effectiveness in data selection. These baselines included: (1) **SelectIT** (Liu et al., 2024), which selects data using model feedback at the token, sentence, and model levels to identify useful samples; (2) **LESS** (Xia et al., 2024), which leverages LoRA approximated gradient-based influence estimation to prioritize impactful data points; (3) **Random**, which selects a fixed percentage (x%) of the dataset randomly, providing a benchmark for non-strategic selection; (4) **DELIFT with Sentence Embedding Features (SE)**, which uses DELIFT but substitutes sentence embeddings as the feature space, employing a model-independent, pairwise similarity kernel instead of the utility kernel for submodular optimization; and (5) **Full Data**, where the entire dataset is used for fine-tuning, serving as an upper benchmark for performance. For In-Context Learning (ICL), the selected subsets from each baseline were used as the pool of examples, allowing us to evaluate how effectively each method supports ICL by providing relevant and informative data.

Model	Qwen2						Phi-3					
Method	ICL			QLoRA			ICL			QLoRA		
	ROUGE	BGE	LAJ	ROUGE	BGE	LAJ	ROUGE	BGE	LAJ	ROUGE	BGE	LAJ
Initial	44.11	70.49	2.43	48.49	80.85	2.62	40.66	58.68	1.52	43.96	69.56	2.29
Random	55.57	85.26	2.91	55.52	85.53	2.94	45.76	76.19	2.45	58.94	82.41	2.89
SelectIT	63.07	86.38	3.18	65.42	87.50	3.20	63.49	85.27	2.96	64.09	85.07	3.16
LESS	64.28	85.41	3.29	69.85	89.33	3.45	66.01	87.20	3.19	67.53	88.17	3.22
DELIFT (SE)	61.07	85.16	3.45	74.05	**92.47**	3.58	68.84	88.46	3.32	69.30	88.62	3.35
DELIFT	**69.49**	**87.94**	**3.60**	**74.19**	92.23	**3.65**	**74.11**	**89.41**	**3.57**	**74.38**	**91.55**	**3.57**
Full Data	*66.08*	*87.84*	*3.65*	*76.83*	*92.63*	*3.74*	*71.23*	*91.10*	*3.52*	*77.12*	*91.10*	*3.64*

Table 5: Results on Use Case 3: IBM and Government. **Bold** indicates the best performance. There is *only* a 0.31% performance percentage drop from Full Data to DELIFT after pruning 70% of the data, and a 3.89% performance percentage drop from DELIFT to the next best baseline.

Model	Qwen2						Phi-3					
Method	ICL			QLoRA			ICL			QLoRA		
	ROUGE	BGE	LAJ	ROUGE	BGE	LAJ	ROUGE	BGE	LAJ	ROUGE	BGE	LAJ
Initial	51.51	66.97	1.77	54.18	78.27	2.50	40.42	58.23	1.26	40.94	58.12	1.29
Random	54.38	79.12	2.57	59.23	82.02	2.66	44.29	59.45	1.33	50.29	61.52	1.60
SelectIT	58.03	83.75	2.82	63.26	84.01	2.87	47.35	74.15	2.54	56.88	80.47	2.70
LESS	67.16	85.76	2.94	69.72	86.63	3.26	60.97	81.41	2.84	61.56	81.53	2.88
DELIFT (SE)	73.75	88.01	3.26	74.84	88.79	3.30	64.44	83.95	3.03	66.35	84.77	3.14
DELIFT	**76.94**	**90.41**	**3.33**	**77.56**	**89.99**	**3.34**	**66.55**	**84.65**	**3.25**	**67.09**	**85.17**	**3.32**
Full Data	*77.78*	*90.31*	*3.35*	*78.72*	*90.77*	*3.48*	*68.47*	*85.93*	*3.33*	*70.48*	*86.06*	*3.44*

Table 6: Results on Use Case 3: SQuAD and HotpotQA. **Bold** indicates the best performance. There is *only* a 1.94% performance percentage drop from Full Data to DELIFT after pruning 70% of the data, and a 2.78% performance percentage drop from DELIFT to the next best baseline.

4.3 Results and Analysis

To ensure a fair and comprehensive evaluation of DELIFT, we conducted experiments across three distinct fine-tuning scenarios: instruction tuning, task-specific fine-tuning, and continual fine-tuning. For all subset selection methods—including DELIFT, Random, SelectIT, LESS, and DELIFT with Sentence Embdedding Features (SE)—we consistently selected 30% of the dataset as a subset, enabling direct comparisons between methods and with the full dataset baseline (see Section 4.4 for an ablation study examining the impact of subset size).

Use Case 1: Instruction Tuning Our first set of experiments focused on instruction tuning, a crucial task to enhancing a model's ability to follow diverse instructions. As shown in Tables 1 and 2, DELIFT achieved a minimal performance drop of only 5.60% compared to using the full dataset while reducing the dataset by 70%. This demonstrates DELIFT's capability to retain the most informative samples essential for instruction tuning. Furthermore, DELIFT outperformed other subset selection methods, achieving a 2.74% improvement and a substantial 26.21% advantage over the next best and worst-performing baselines, respectively. These results underscore DELIFT's superior ability to maintain high performance with significantly reduced data, highlighting its efficacy in instruction tuning.

Use Case 2: Task-Specific Fine-Tuning In the task-specific fine-tuning scenario, we evaluated DELIFT using two dataset pairs: (1) HotpotQA (Yang et al., 2018) with MMLU (Hendrycks et al., 2021), and (2) MixInstruct paired with MT-Bench (Zheng et al., 2023). Results, presented in Tables 4 and 3, demonstrate DELIFT's consistent and competitive performance across different task pairs. A particularly noteworthy outcome emerged from the HotpotQA-MMLU pair, where DELIFT not only matched but exceeded the performance of the full dataset, achieving a 5.51% improvement. This indicates that DELIFT's selective approach can effectively filter out noise and focus on the most relevant and informative samples, yielding enhanced task-specific adaptation even with reduced data.

Use Case 3: Continual Fine-Tuning The third use case examined DELIFT's efficacy in continual fine-tuning, where models need to incorporate new information while retaining previously learned knowledge. As detailed in Tables 5 and 6, DELIFT demonstrated remarkable consistency, showing only a marginal 1.13% performance drop compared to using the full dataset. Moreover, DELIFT outperformed the second-best baseline by 3.33% and the worst baseline by 23.88%, highlighting its superiority in data selection. In specialized tasks such as query rewriting, DELIFT even surpassed the performance of the full dataset, suggesting that its selective approach effectively prunes noisy or irrelevant data points, thereby enhancing model performance.

4.4 Ablation Study: Impact of Subset Size

To assess how subset size influences DELIFT's performance, we conducted an ablation study by varying the subset size from 5% to 100% of the full dataset across three use cases. The results, detailed in Appendix B and illustrated in Figure 2, show that LAJ scores generally increase with subset size. Utilizing the full dataset consistently yields the highest performance, highlighting the benefits of larger training sets. However, for methods such as DELIFT, SelectIT, and LESS, performance gains plateau or slow beyond a 50% subset size, indicating that additional data offers minimal benefits and may introduce redundancy. Importantly, DELIFT outperforms all baselines across subset sizes from 5% to 100%, demonstrating its robustness and effectiveness in selecting informative samples regardless of subset size. These findings suggest that carefully selected smaller datasets can achieve comparable performance to larger, unfiltered datasets, which is particularly valuable for resource-intensive large language models.

4.5 Discussion

The comprehensive results across all three use cases highlight DELIFT's effectiveness and versatility. By consistently reducing data requirements by up to 70% while maintaining—and in some cases improving—performance, DELIFT addresses a critical challenge in large language model fine-tuning. The superior performance of DELIFT can be attributed to its novel pairwise utility metric and the use of tailored submodular functions for each fine-tuning stage. This approach enables DELIFT to select not only representative and diverse samples but also to reduce noise present in the full dataset. The ability to outperform full datasets in certain scenarios, particularly in niche tasks like query rewriting, underscores DELIFT's capacity to *distill the most relevant and informative data points*. These findings have significant implications for the accessibility and efficiency of LLM fine-tuning. By dramatically reducing the amount of data required for effective fine-tuning, DELIFT paves the way for more widespread adoption and application of large language models across various domains, especially in resource-constrained environments. Furthermore, DELIFT's consistent outperformance of existing data selection techniques across various fine-tuning scenarios and model scales demonstrates its robustness and broad applicability, making it a valuable tool for researchers and practitioners alike. In conclusion, our experimental results firmly establish DELIFT as a powerful and efficient method for data selection in LLM fine-tuning. By addressing the critical challenge of optimal data selection, DELIFT not only enhances the efficiency of model training but also opens new possibilities for fine-tuning large language models in domains where data or computational resources may be limited.

5 Conclusion, Limitations, and Future Work

In this paper, we introduced DELIFT, a novel approach to data-efficient fine-tuning of large language models by employing a versatile pairwise utility metric combined with submodular optimization techniques for optimal data selection. Empirical evaluations showed that DELIFT can reduce data and computational requirements by up to 70% while achieving performance comparable to the full dataset, and outperforming existing data selection methods by up to 26% in effectiveness. These results suggest that DELIFT offers a promising method for improving the accessibility of LLM adaptation, especially for resource-constrained scenarios. However, our approach has limitations, including potential sensitivity to the quality and diversity of initial data and the risk of bias amplification inherent in the selected data. Future work will explore integrating DELIFT with data augmentation techniques to improve robustness, incorporating fairness constraints to mitigate biases, and extending the approach to emerging model architectures and multimodal learning. Our ongoing efforts are directed toward ensuring that DELIFT contributes to responsible and equitable AI development while maximizing efficiency.

REFERENCES

Marah Abdin, Jyoti Aneja, Hany Awadalla, Ahmed Awadallah, Ammar Ahmad Awan, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Jianmin Bao, Harkirat Behl, Alon Benhaim, Misha Bilenko, Johan Bjorck, Sébastien Bubeck, Martin Cai, Qin Cai, Vishrav Chaudhary, Dong Chen, Dongdong Chen, Weizhu Chen, Yen-Chun Chen, Yi-Ling Chen, Hao Cheng, Parul Chopra, Xiyang Dai, Matthew Dixon, Ronen Eldan, Victor Fragoso, Jianfeng Gao, Mei Gao, Min Gao, Amit Garg, Allie Del Giorno, Abhishek Goswami, Suriya Gunasekar, Emman Haider, Junheng Hao, Russell J. Hewett, Wenxiang Hu, Jamie Huynh, Dan Iter, Sam Ade Jacobs, Mojan Javaheripi, Xin Jin, Nikos Karampatziakis, Piero Kauffmann, Mahoud Khademi, Dongwoo Kim, Young Jin Kim, Lev Kurilenko, James R. Lee, Yin Tat Lee, Yuanzhi Li, Yunsheng Li, Chen Liang, Lars Liden, Xihui Lin, Zeqi Lin, Ce Liu, Liyuan Liu, Mengchen Liu, Weishung Liu, Xiaodong Liu, Chong Luo, Piyush Madan, Ali Mahmoudzadeh, David Majercak, Matt Mazzola, Caio César Teodoro Mendes, Arindam Mitra, Hardik Modi, Anh Nguyen, Brandon Norick, Barun Patra, Daniel Perez-Becker, Thomas Portet, Reid Pryzant, Heyang Qin, Marko Radmilac, Liliang Ren, Gustavo de Rosa, Corby Rosset, Sambudha Roy, Olatunji Ruwase, Olli Saarikivi, Amin Saied, Adil Salim, Michael Santacroce, Shital Shah, Ning Shang, Hiteshi Sharma, Yelong Shen, Swadheen Shukla, Xia Song, Masahiro Tanaka, Andrea Tupini, Praneetha Vaddamanu, Chunyu Wang, Guanhua Wang, Lijuan Wang, Shuohang Wang, Xin Wang, Yu Wang, Rachel Ward, Wen Wen, Philipp Witte, Haiping Wu, Xiaoxia Wu, Michael Wyatt, Bin Xiao, Can Xu, Jiahang Xu, Weijian Xu, Jilong Xue, Sonali Yadav, Fan Yang, Jianwei Yang, Yifan Yang, Ziyi Yang, Donghan Yu, Lu Yuan, Chenruidong Zhang, Cyril Zhang, Jianwen Zhang, Li Lyna Zhang, Yi Zhang, Yue Zhang, Yunan Zhang, and Xiren Zhou. Phi-3 technical report: A highly capable language model locally on your phone, 2024. URL https://arxiv.org/abs/2404.14219.

Gantavya Bhatt, Yifang Chen, Arnav Das, Jifan Zhang, Sang Truong, Stephen Mussmann, Yinglun Zhu, Jeff Bilmes, Simon Du, Kevin Jamieson, Jordan Ash, and Robert Nowak. An experimental design framework for label-efficient supervised finetuning of large language models. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Findings of the Association for Computational Linguistics ACL 2024*, pp. 6549–6560, Bangkok, Thailand and virtual meeting, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-acl.390. URL https://aclanthology.org/2024.findings-acl.390.

Jeff Bilmes. Submodularity in machine learning and artificial intelligence, 2022. URL https://arxiv.org/abs/2202.00132.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 1877–1901. Curran Associates, Inc., 2020a. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners, 2020b. URL https://arxiv.org/abs/2005.14165.

Alexander Bukharin and Tuo Zhao. Data diversity matters for robust instruction tuning, 2024. URL https://arxiv.org/abs/2311.14736.

Lichang Chen, Shiyang Li, Jun Yan, Hai Wang, Kalpa Gunaratna, Vikas Yadav, Zheng Tang, Vijay Srinivasan, Tianyi Zhou, Heng Huang, and Hongxia Jin. Alpagasus: Training a better alpaca with fewer data, 2024. URL https://arxiv.org/abs/2307.08701.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems, 2021. URL https://arxiv.org/abs/2110.14168.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In Jill Burstein, Christy Doran, and Thamar Solorio (eds.), *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423. URL https://aclanthology.org/N19-1423.

Qianlong Du, Chengqing Zong, and Jiajun Zhang. Mods: Model-oriented data selection for instruction tuning, 2023. URL https://arxiv.org/abs/2311.15653.

Satoru Fujishige. *Submodular functions and optimization*. Elsevier, 2005.

Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. Don't stop pretraining: Adapt language models to domains and tasks. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault (eds.), *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 8342–8360, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.740. URL https://aclanthology.org/2020.acl-main.740.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. *Proceedings of the International Conference on Learning Representations (ICLR)*, 2021.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, Tom Hennigan, Eric Noland, Katie Millican, George van den Driessche, Bogdan Damoc, Aurelia Guy, Simon Osindero, Karen Simonyan, Erich Elsen, Oriol Vinyals, Jack W. Rae, and Laurent Sifre. Training compute-optimal large language models. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, Red Hook, NY, USA, 2024. Curran Associates Inc. ISBN 9781713871088.

Rishabh Iyer, Ninad Khargoankar, Jeff Bilmes, and Himanshu Asanani. Submodular combinatorial information measures with applications in machine learning. In Vitaly Feldman, Katrina Ligett, and Sivan Sabato (eds.), *Proceedings of the 32nd International Conference on Algorithmic Learning Theory*, volume 132 of *Proceedings of Machine Learning Research*, pp. 722–754. PMLR, 16–19 Mar 2021. URL https://proceedings.mlr.press/v132/iyer21a.html.

Dongfu Jiang, Xiang Ren, and Bill Yuchen Lin. Llm-blender: Ensembling large language models with pairwise ranking and generative fusion. (arXiv:2306.02561), June 2023. URL http://arxiv.org/abs/2306.02561. arXiv:2306.02561 [cs].

Krishnateja Killamsetty, Durga Sivasubramanian, Ganesh Ramakrishnan, Abir De, and Rishabh Iyer. Grad-match: Gradient matching based data subset selection for efficient deep model training, 2021a. URL https://arxiv.org/abs/2103.00123.

Krishnateja Killamsetty, Durga Sivasubramanian, Ganesh Ramakrishnan, and Rishabh Iyer. Glister: Generalization based data subset selection for efficient and robust learning, 2021b. URL https://arxiv.org/abs/2012.10630.

Krishnateja Killamsetty, Alexandre V. Evfimievski, Tejaswini Pedapati, Kiran Kate, Lucian Popa, and Rishabh Iyer. Milo: Model-agnostic subset selection framework for efficient model training and tuning, 2023. URL https://arxiv.org/abs/2301.13287.

Seungone Kim, Jamin Shin, Yejin Cho, Joel Jang, Shayne Longpre, Hwaran Lee, Sangdoo Yun, Seongjin Shin, Sungdong Kim, James Thorne, et al. Prometheus: Inducing fine-grained evaluation capability in language models. *arXiv preprint arXiv:2310.08491*, 2023.

Ming Li, Yong Zhang, Zhitao Li, Jiuhai Chen, Lichang Chen, Ning Cheng, Jianzong Wang, Tianyi Zhou, and Jing Xiao. From quantity to quality: Boosting LLM performance with self-guided data selection for instruction tuning. In Kevin Duh, Helena Gomez, and Steven Bethard (eds.), *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 7595–7628, Mexico City, Mexico, June 2024. Association for Computational Linguistics. URL https://aclanthology.org/2024.naacl-long.421.

Chin-Yew Lin. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pp. 74–81, Barcelona, Spain, July 2004. Association for Computational Linguistics. URL https://aclanthology.org/W04-1013.

Liangxin Liu, Xuebo Liu, Derek F. Wong, Dongfang Li, Ziyi Wang, Baotian Hu, and Min Zhang. Selectit: Selective instruction tuning for large language models via uncertainty-aware self-reflection, 2024. URL https://arxiv.org/abs/2402.16705.

Shayne Longpre, Le Hou, Tu Vu, Albert Webson, Hyung Won Chung, Yi Tay, Denny Zhou, Quoc V. Le, Barret Zoph, Jason Wei, and Adam Roberts. The flan collection: designing data and methods for effective instruction tuning. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org, 2023.

Andrea Madotto, Zhaojiang Lin, Zhenpeng Zhou, Seungwhan Moon, Paul Crook, Bing Liu, Zhou Yu, Eunjoon Cho, Pascale Fung, and Zhiguang Wang. Continual learning in task-oriented dialogue systems. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih (eds.), *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 7452–7467, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.590. URL https://aclanthology.org/2021.emnlp-main.590.

Baharan Mirzasoleiman, Jeff Bilmes, and Jure Leskovec. Coresets for data-efficient training of machine learning models, 2020. URL https://arxiv.org/abs/1906.01827.

Swaroop Mishra, Daniel Khashabi, Chitta Baral, and Hannaneh Hajishirzi. Cross-task generalization via natural language crowdsourcing instructions. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio (eds.), *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3470–3487, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-long.244. URL https://aclanthology.org/2022.acl-long.244.

George L. Nemhauser, Laurence A. Wolsey, and Marshall L. Fisher. An analysis of approximations for maximizing submodular set functions—i. *Mathematical Programming*, 14:265–294, 1978. URL https://api.semanticscholar.org/CorpusID:206800425.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 27730–27744. Curran Associates, Inc., 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/b1efde53be364a73914f58805a001731-Paper-Conference.pdf.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. SQuAD: 100,000+ questions for machine comprehension of text. In Jian Su, Kevin Duh, and Xavier Carreras (eds.), *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 2383–2392, Austin, Texas, November 2016. Association for Computational Linguistics. doi: 10.18653/v1/D16-1264. URL https://aclanthology.org/D16-1264.

Victor Sanh, Albert Webson, Colin Raffel, Stephen H. Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Teven Le Scao, Arun Raja, Manan Dey, M Saiful Bari, Canwen Xu, Urmish Thakker, Shanya Sharma Sharma, Eliza Szczechla, Taewoon Kim, Gunjan Chhablani, Nihal Nayak, Debajyoti Datta, Jonathan Chang, Mike Tian-Jian Jiang, Han Wang, Matteo Manica, Sheng Shen, Zheng Xin Yong, Harshit Pandey, Rachel Bawden, Thomas Wang, Trishala Neeraj, Jos Rozen, Abheesht Sharma, Andrea Santilli, Thibault Fevry, Jason Alan Fries, Ryan Teehan, Stella Biderman, Leo Gao, Tali Bers, Thomas Wolf, and Alexander M. Rush. Multitask prompted training enables zero-shot task generalization, 2021.

Ben Sorscher, Robert Geirhos, Shashank Shekhar, Surya Ganguli, and Ari S. Morcos. Beyond neural scaling laws: beating power law scaling via data pruning, 2023. URL https://arxiv.org/abs/2206.14486.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. Llama: Open and efficient foundation language models, 2023. URL https://arxiv.org/abs/2302.13971.

Jason Wei, Maarten Bosma, Vincent Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V Le. Finetuned language models are zero-shot learners. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=gEZrGCozdqR.

Ronald J. Williams and David Zipser. A learning algorithm for continually running fully recurrent neural networks. *Neural Computation*, 1(2):270–280, 1989. doi: 10.1162/neco.1989.1.2.270.

Tongtong Wu, Linhao Luo, Yuan-Fang Li, Shirui Pan, Thuy-Trang Vu, and Gholamreza Haffari. Continual learning for large language models: A survey, 2024. URL https://arxiv.org/abs/2402.01364.

Mengzhou Xia, Sadhika Malladi, Suchin Gururangan, Sanjeev Arora, and Danqi Chen. LESS: Selecting influential data for targeted instruction tuning. In *International Conference on Machine Learning (ICML)*, 2024.

Shitao Xiao, Zheng Liu, Peitian Zhang, and Niklas Muennighoff. C-pack: Packaged resources to advance general chinese embedding, 2023.

Tianci Xue, Ziqi Wang, Yixia Li, Yun Chen, and Guanhua Chen. PACIT: Unlocking the power of examples for better in-context instruction tuning. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Findings of the Association for Computational Linguistics ACL 2024*, pp. 654–665, Bangkok, Thailand and virtual meeting, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-acl.36. URL https://aclanthology.org/2024.findings-acl.36.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong Tang, Jialin Wang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Ma, Jianxin Yang, Jin Xu, Jingren Zhou, Jinze Bai, Jinzheng He, Junyang Lin, Kai Dang, Keming Lu, Keqin Chen, Kexin Yang, Mei Li, Mingfeng Xue, Na Ni, Pei Zhang, Peng Wang, Ru Peng, Rui Men, Ruize Gao, Runji Lin, Shijie Wang, Shuai Bai, Sinan Tan, Tianhang Zhu, Tianhao Li, Tianyu Liu, Wenbin Ge, Xiaodong Deng, Xiaohuan Zhou, Xingzhang Ren, Xinyu Zhang, Xipin Wei, Xuancheng Ren, Xuejing Liu, Yang Fan, Yang Yao, Yichang Zhang, Yu Wan, Yunfei Chu, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, Zhifang Guo, and Zhihao Fan. Qwen2 technical report, 2024. URL https://arxiv.org/abs/2407.10671.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. Hotpotqa: A dataset for diverse, explainable multi-hop question answering, 2018. URL https://arxiv.org/abs/1809.09600.

Jipeng Zhang, Yaxuan Qin, Renjie Pi, Weizhong Zhang, Rui Pan, and Tong Zhang. Tagcos: Task-agnostic gradient clustered coreset selection for instruction tuning data, 2024. URL https://arxiv.org/abs/2407.15235.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric. P Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. Judging llm-as-a-judge with mt-bench and chatbot arena, 2023.

Chunting Zhou, Pengfei Liu, Puxin Xu, Srinivasan Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, LILI YU, Susan Zhang, Gargi Ghosh, Mike Lewis, Luke Zettlemoyer, and Omer Levy. Lima: Less is more for alignment. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 55006–55021. Curran Associates, Inc., 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/ac662d74829e4407ce1d126477f4a03a-Paper-Conference.pdf.

Appendix

A Theoretical Foundations and Connections Between the Utility Metric and Information Theory

Theorem 1 *Let $y_i = (y_{i1}, y_{i2}, \ldots, y_{iT})$ be a sequence of tokens with ground truth distribution GT_i, where GT_i assigns probability 1 to the sequence y_i and 0 to all other sequences. Let $p(y_i \mid x_i)$ be the predicted probability of y_i given input x_i, and $p(y_i \mid x_i, x_j, y_j)$ be the predicted probability of y_i given x_i and an in-context example (x_j, y_j). Define the utility metric UF_{ij} using a general distance metric $d(\cdot, \cdot)$ between probability distributions:*

$$UF_{ij} = d(GT_i,\, p(y_i \mid x_i)) - d(GT_i,\, p(y_i \mid x_i, x_j, y_j)).$$

Claim: *When the distance metric $d(\cdot, \cdot)$ is the Kullback-Leibler divergence D_{KL}, the utility metric UF_{ij} is equal to the pointwise mutual information (PMI) between the sequence y_i and the in-context example (x_j, y_j) conditioned on x_i:*

$$UF_{ij} = \mathrm{PMI}(y_i;\, x_j, y_j \mid x_i) = \log \frac{p(y_i \mid x_i, x_j, y_j)}{p(y_i \mid x_i)}.$$

Furthermore, UF_{ij} can be expressed as the sum of conditional PMI over the tokens in y_i:

$$UF_{ij} = \sum_{t=1}^{T} \mathrm{PMI}\left(y_{it};\, x_j, y_j \mid x_i, y_{i,<t}\right),$$

where $y_{i,<t} = (y_{i1}, y_{i2}, \ldots, y_{i(t-1)})$ denotes the sequence of previous tokens up to position $t-1$.

Proof:

1. Computing KL-Divergence Between Ground Truth and Predicted Distributions:

Since GT_i assigns probability 1 to the specific sequence y_i, the KL-divergence simplifies as follows:

$$d(GT_i,\, p(y_i \mid \cdot)) = D_{KL}(GT_i \,\|\, p(y_i \mid \cdot)) = -\log p(y_i \mid \cdot),$$

because the KL-divergence between a one-hot distribution and any other distribution reduces to the negative log-probability of the assigned event.

2. Computing the Utility Metric UF_{ij}:

The utility metric becomes:

$$\begin{aligned} UF_{ij} &= -\log p(y_i \mid x_i) + \log p(y_i \mid x_i, x_j, y_j) \\ &= \log \frac{p(y_i \mid x_i, x_j, y_j)}{p(y_i \mid x_i)}. \end{aligned}$$

3. Expressing UF_{ij} as Pointwise Mutual Information:

The conditional pointwise mutual information between y_i and (x_j, y_j) given x_i is defined as:

$$\mathrm{PMI}(y_i;\, x_j, y_j \mid x_i) = \log \frac{p(y_i, x_j, y_j \mid x_i)}{p(y_i \mid x_i)\, p(x_j, y_j \mid x_i)}.$$

Using the chain rule:

$$p(y_i, x_j, y_j \mid x_i) = p(y_i \mid x_i, x_j, y_j)\, p(x_j, y_j \mid x_i).$$

Substituting back:

$$\begin{aligned}\mathrm{PMI}(y_i;\, x_j, y_j \mid x_i) &= \log \frac{p(y_i \mid x_i, x_j, y_j)\, p(x_j, y_j \mid x_i)}{p(y_i \mid x_i)\, p(x_j, y_j \mid x_i)} \\ &= \log \frac{p(y_i \mid x_i, x_j, y_j)}{p(y_i \mid x_i)}.\end{aligned}$$

Therefore:

$$UF_{ij} = \mathrm{PMI}(y_i;\, x_j, y_j \mid x_i).$$

4. Expanding UF_{ij} as Sum of Conditional PMI Terms:

We expand $p(y_i \mid \cdot)$ using the chain rule:

$$p(y_i \mid \cdot) = \prod_{t=1}^{T} p(y_{it} \mid \cdot, y_{i,<t}),$$

where $y_{i,<t}$ is the sequence of previous tokens up to time $t-1$.

Substituting back into UF_{ij}:

$$\begin{aligned}UF_{ij} &= \log \frac{\prod_{t=1}^{T} p(y_{it} \mid x_i, x_j, y_j, y_{i,<t})}{\prod_{t=1}^{T} p(y_{it} \mid x_i, y_{i,<t})} \\ &= \sum_{t=1}^{T} \left[\log p(y_{it} \mid x_i, x_j, y_j, y_{i,<t}) - \log p(y_{it} \mid x_i, y_{i,<t})\right] \\ &= \sum_{t=1}^{T} \mathrm{PMI}\left(y_{it};\, x_j, y_j \mid x_i, y_{i,<t}\right).\end{aligned}$$

This shows that UF_{ij} is the sum of the conditional PMI of each token y_{it} with (x_j, y_j) given x_i and the previous tokens.

Conclusion:

When $d(\cdot,\cdot) = D_{KL}$, the utility metric UF_{ij} precisely equals the conditional PMI between y_i and (x_j, y_j) given x_i.

A.1 Why Euclidean Distance is Preferred Over KL-Divergence for Subset Selection

The effectiveness of subset selection algorithms, including facility location functions, depends critically on the properties of the chosen distance metric d(·,·). Euclidean distance offers several key advantages over KL-divergence for this purpose:

1. **Mathematical Properties**
   - Euclidean distance is non-negative, finite, and symmetric (d(a,b) = d(b,a))
   - KL-divergence can be infinite or undefined with zero probabilities and lacks symmetry $D_{KL}(P \parallel Q) \neq D_{KL}(Q \parallel P)$
2. **Computational Advantages**
   - Euclidean distance uses simple arithmetic operations (subtraction, squares, square roots)
   - KL-divergence requires more complex logarithmic calculations and division operations

3. **Robustness in Practice**
    - Euclidean distance handles zero probabilities gracefully
    - KL-divergence becomes undefined with zero probabilities, which occur frequently in real data

Impact on Subset Selection: The facility location function requires positive, finite similarity measures to model coverage effects accurately. Euclidean distance satisfies these requirements, while KL-divergence's potential negative or infinite values can disrupt optimization.

Conclusion: While KL-divergence offers theoretical connections to mutual information, Euclidean distance provides:

- Guaranteed positive and finite utility metrics
- Superior computational efficiency
- Better numerical stability

These practical advantages make Euclidean distance the preferred choice for computing the utility metric UF_{ij} in subset selection algorithms.

B Subset Size Comparison

To assess how subset size influences the performance of DELIFT, we performed an ablation study by varying the subset size from 5% to 100% (specifically 5%, 15%, 30%, 50%, 100%) of the entire dataset across three distinct use cases. Figure 2 illustrates the performance metric LAJ as a function of subset size for each fine-tuning scenario.

B.1 General Observations

- **Performance Increases with Subset Size:** Across all methods, LAJ scores generally improve as the subset size increases. Utilizing the full dataset consistently yields the highest performance, underscoring the benefits of a larger training set.
- **Diminishing Returns Beyond 50%:** For methods such as DELIFT, SelectIT, and LESS, performance gains plateau or slow down beyond a 50% subset size. This suggests that additional data beyond this point offers minimal benefits and may introduce redundancy.

B.2 Detailed Analysis of Methods

B.2.1 Initial vs. Random Selection

- **Initial Baseline:** Consistently records the lowest scores across all subset sizes, indicating that models without data-informed selection struggle to generate quality responses.
- **Random Selection:** Slightly outperforms the Initial baseline but maintains a relatively flat performance curve. This lack of significant improvement highlights that uninformed data selection does not substantially enhance model quality.

B.2.2 SelectIT and LESS Methods

- **LESS:** Demonstrates a strong upward trend, particularly when subset sizes increase from 15% to 50%. This indicates that LESS effectively selects informative subsets, especially in the mid-range subset sizes, but is sub-optimal with smaller subset sizes.
- **SelectIT:** Initially lags behind DELIFT and LESS but shows steady improvement with larger subset sizes. For subset sizes above 50%, SelectIT approaches the performance of DELIFT, suggesting its heuristic-driven selection becomes more effective with more data.

B.2.3 DELIFT Variants

- **DELIFT vs. DELIFT (SE):** DELIFT consistently outperforms DELIFT (SE), which uses sentence embeddings, highlighting the superiority of DELIFT's utility-based kernel in capturing data informativeness.

- **DELIFT vs. Other Methods:** DELIFT outperforms all other subset selection methods across all subset sizes, particularly below 50%. This effectiveness is attributed to DELIFT's strategy of identifying the most informative samples early on, making it ideal for scenarios with limited computational resources.
- **DELIFT vs. Full Data:** At smaller subset sizes (e.g., 15%, 30%), DELIFT achieves LAJ scores close to the Full Data baseline. In ICL fine-tuning scenarios, a 30% subset size with DELIFT nearly matches Full Data performance, demonstrating its efficiency in data reduction without significant loss in performance.

B.3 Impact on Different Fine-Tuning Scenarios

- **ICL vs. QLoRA:** QLoRA fine-tuning generally yields higher scores than ICL across all methods, suggesting that QLoRA benefits more from effective data selection strategies. DELIFT, in particular, shows more pronounced improvements in QLoRA settings, indicating its subsets are well-suited for efficient parameter tuning.
- **Use Case Comparisons:** In Use Case 3 (IBM and Government datasets), DELIFT achieves the highest gains relative to the Initial baseline across both ICL and QLoRA scenarios. This effectiveness is likely due to the nature of query rewriting tasks, where DELIFT's informed data selection effectively eliminates redundant or irrelevant examples, resulting in a higher-quality training set.

C Prometheus Rubric

The Prometheus model served as an LLM-as-a-Judge to evaluate response quality from different data selection methods. Table 7 contains the general rubric used for the Prometheus model scoring on all use cases and settings (except for the experiments on the query-rewriting task using the IBM-proprietary data).

C.1 Usage Notes

- Each response is evaluated independently based on the criteria above.
- The cumulative score reflects the overall quality and effectiveness of the response.
- Final LAJ scores are obtained by averaging the scores across all criteria.

D LLM-as-Judges Scores

In Tables 8 and 9, we show the distribution of Prometheus scores on one particular setting: Use Case 1, MixInstruct training and MixInstruct validation sets on the Qwen2-72B-Instruct model. These figures make clear that the average LGA scores computed in Tables 1-6 are true averages of a distribution of scores, not averages of a combination of just 1's and 5's.

D.1 Interpretation of Score Distributions

D.1.1 Overall Trends

- **Score Variability:** There is significant variability in score distributions across different methods. The Initial and Random baselines show a concentration of scores between 2.5 and 3.5, indicating average to subpar performance.
- **Enhanced Performance with Advanced Methods:** Methods like SelectIT, LESS, DELIFT (SE), and DELIFT exhibit score distributions skewed towards higher values (3.5 to 4.0), with DELIFT showing the highest concentration above 3.5. This highlights their effectiveness in selecting informative and useful data for fine-tuning.

D.1.2 Method-Specific Observations

- **Initial and Random Methods:** Both methods have lower mean scores (around 3.0 to 3.2) with wide spreads, suggesting inconsistent and generally lower-quality responses.

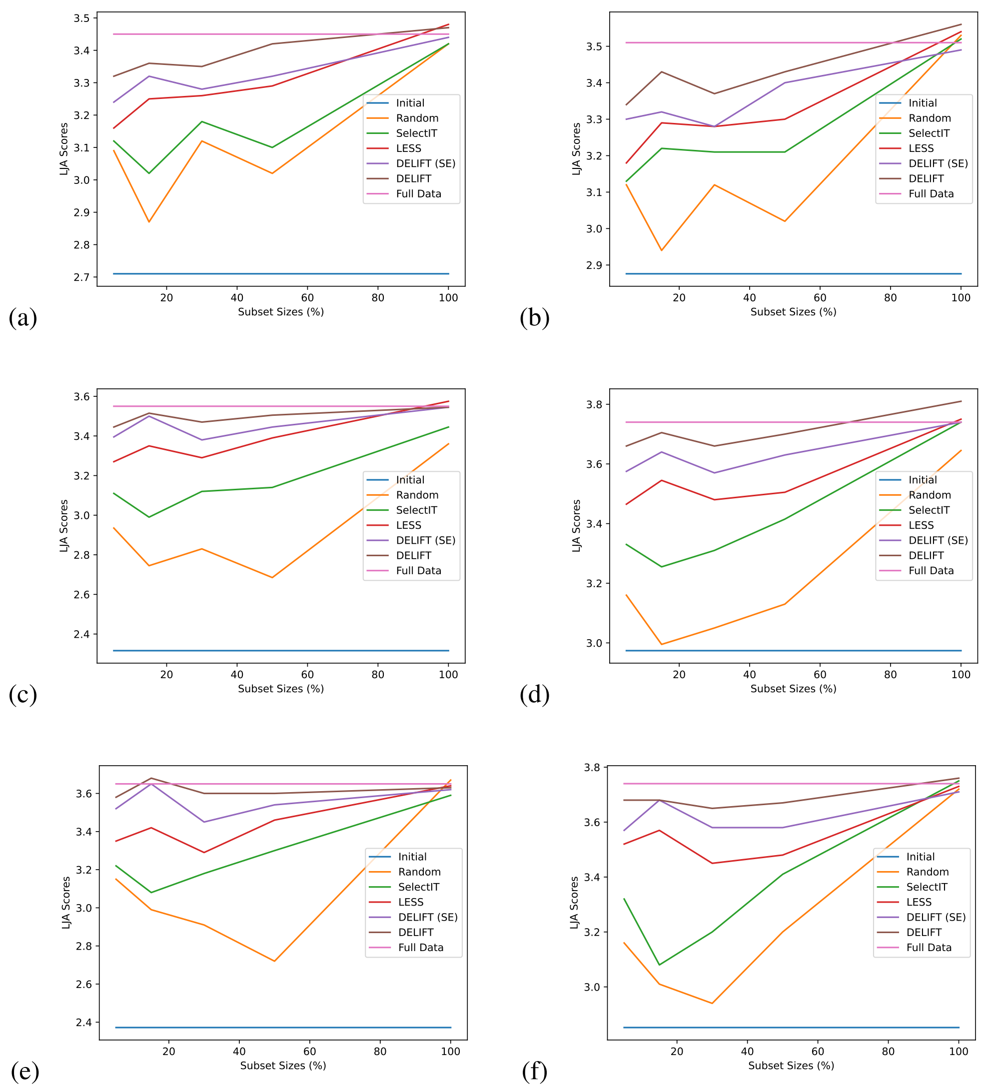


Figure 2: Graphs of LLM-A-J scores (y-axis) of Qwen2-72B-Instruct with varying subset sizes (x-axis) of Use Case 1 on MixInstruct for **(a)** ICL and **(b)** QLoRA, Use Case 2 on MixInstruct and MT-Bench for **(c)** ICL and **(d)** QLoRA, and Use Case 3 on IBM and Government for **(e)** ICL and **(f)** QLoRA.

Evaluate the model's ability to follow instructions and deliver a high-quality response across the following dimensions: 1. **Instruction Following**: How accurately and fully does the model adhere to the given instruction? 2. **Accuracy**: Is the information correct, reliable, and factually sound? 3. **Relevance**: Does the response directly address the question or task without unnecessary information? 4. **Completeness**: Does the response cover all essential aspects of the instruction or question 5. **Depth**: How thoroughly does the response explore the topic? Does it demonstrate insightful analysis where appropriate? 6. **Clarity**: Is the response well-organized, easy to follow, and free from ambiguity or confusion? 7. **Creativity**: Does the response offer original or innovative approaches where applicable? 8. **Helpfulness**: Does the response effectively meet the user's needs and provide value in solving the problem or addressing the query? **Score of 1**: The response fails to meet expectations across most or all criteria. It does not follow the instruction, contains significant errors or misinformation, lacks relevance, is incomplete or shallow, unclear, unoriginal, and unhelpful. **Score of 2**: "The response shows major deficiencies across several criteria. It partially follows the instruction but includes significant inaccuracies, is often irrelevant, incomplete, or lacks depth, clarity, creativity, and helpfulness. **Score of 3**: "The response is average, meeting some but not all criteria. It follows the instruction but may fall short in terms of accuracy, depth, relevance, or helpfulness. Improvements in clarity and insightfulness may be needed. **Score of 4**: The response is strong, performing well across most criteria. It follows the instruction closely, is mostly accurate and relevant, provides good depth, and is well-structured. Minor improvements could enhance clarity, creativity, or helpfulness. **Score of 5**: "The response excels in all or nearly all criteria. It fully follows the instruction, is highly accurate, directly relevant, complete, and demonstrates depth and insight. The response is well-organized, creative where appropriate, and very helpful in addressing the user's needs.

Table 7: General Prometheus Rubric

- **SelectIT and LESS Methods:**
  - **SelectIT:** Shows improved mean scores, especially in QLoRA settings, indicating its effectiveness in resource-constrained training scenarios.
  - **LESS:** Demonstrates significant performance improvements, with mean scores around 3.26 to 3.28, reflecting effective gradient-based data selection.
- **DELIFT Variants:**
  - **DELIFT (SE):** Skews towards higher scores but not as prominently as DELIFT.
  - **DELIFT:** Achieves the highest average scores (3.35 for ICL and 3.37 for QLoRA), outperforming all other methods and indicating its superior utility-based kernel and submodular optimization.

D.1.3 Comparison with Full Data

- **DELIFT vs. Full Data:** DELIFT nearly matches Full Data performance with only a slight reduction in mean scores (3.35 to 3.37 vs. 3.45 to 3.51). This demonstrates DELIFT's capability to retain most of the model's performance while using significantly less data.
- **Efficiency of Data Pruning:** Full Data shows a modest increase in mean scores compared to DELIFT, but at the cost of substantially higher computational resources. DELIFT offers a more efficient alternative without major sacrifices in performance.

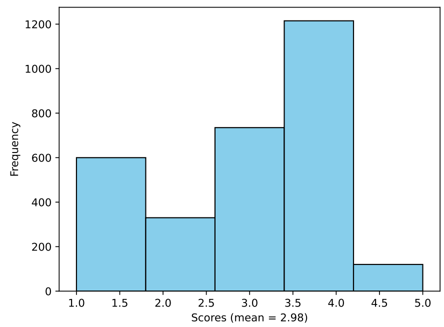
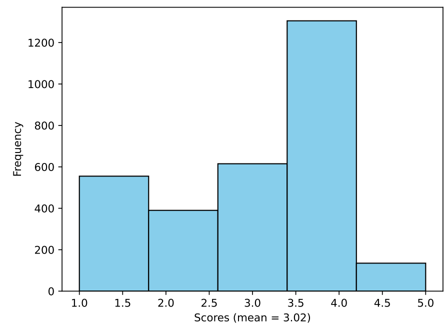
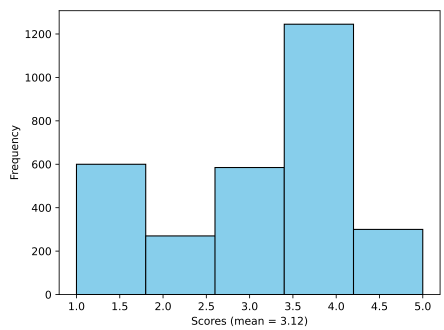
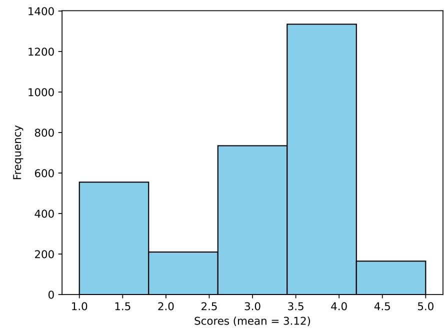
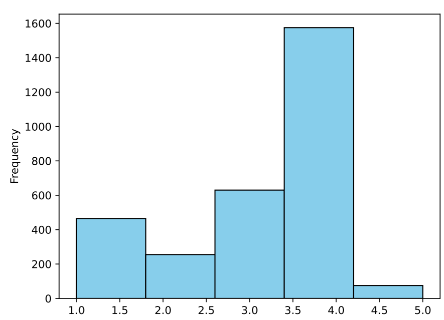
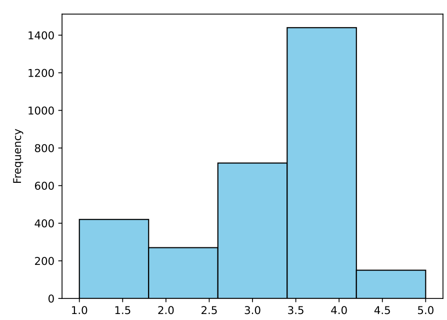
	ICL	QLoRA
Initial		
Random		
SelectIT		

Table 8: LLM-as-Judges score distributions for Use Case 1 with MixInstruct training and validation set on the Qwen2-72B-Instruct model on the Initial, Random, and SelectIT baselines. The corresponding table is Table 1.

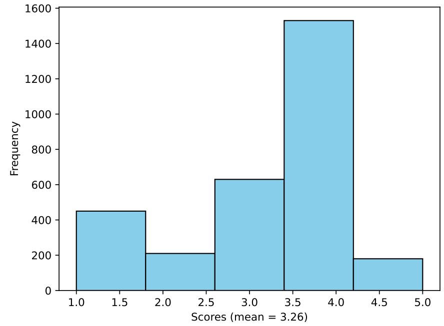
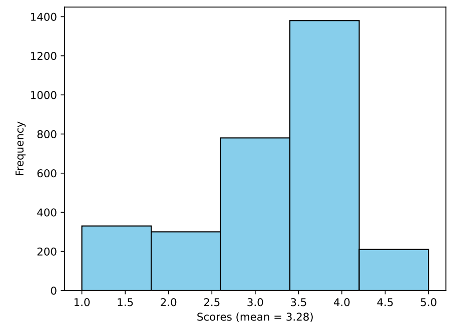
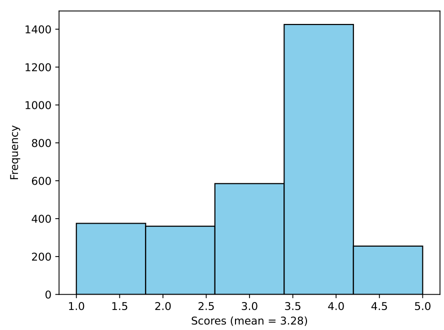
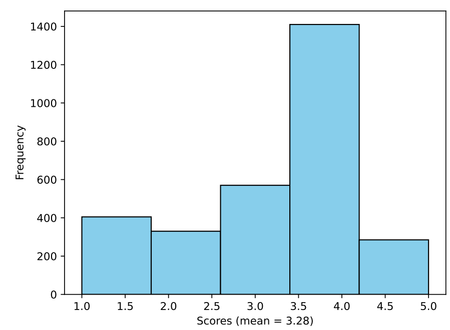
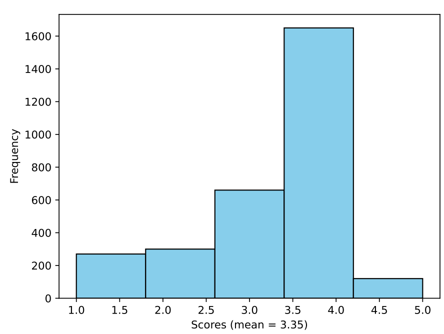
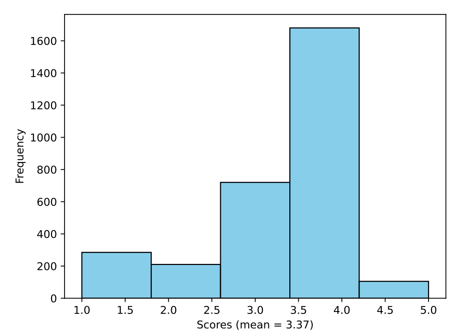
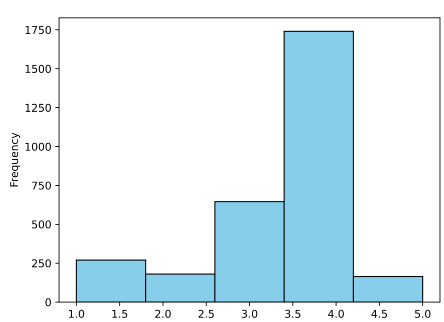
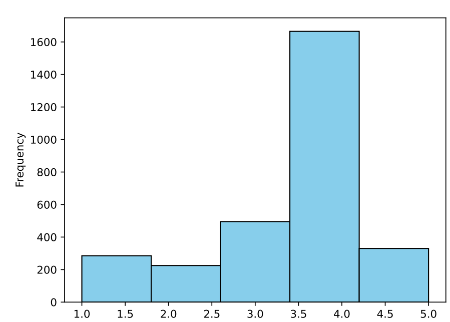
<table>
<tr><th></th><th>ICL</th><th>QLoRA</th></tr>
<tr><td>LESS</td><td></td><td></td></tr>
<tr><td>DELIFT (SE)</td><td></td><td></td></tr>
<tr><td>DELIFT</td><td></td><td></td></tr>
<tr><td>Full Data</td><td></td><td></td></tr>
</table>

Table 9: LLM-as-Judges score distributions for Use Case 1 with MixInstruct training and validation set on the Qwen2-72B-Instruct model on the LESS, DELIFT with Sentence Embedding, DELIFT, and Full Data methods. The corresponding table is Table 1.

E LIMITATIONS

- **Dependence on Initial Data Quality:** DELIFT's effectiveness relies on the diversity and quality of the initial dataset. Biases or lack of diversity in the dataset can propagate to the selected subsets.
- **Scalability Constraints:** While DELIFT is computationally efficient, extremely large datasets may still present challenges in terms of computation and memory.
- **Domain-Specific Performance:** DELIFT's performance may vary across different domains, particularly those requiring specialized knowledge or handling multimodal data.
- **Bias Amplification Risks:** The subset selection process may unintentionally amplify existing biases within the data, necessitating careful mitigation strategies.

F FUTURE WORK

- **Integration with Data Augmentation:** Combining DELIFT with data augmentation techniques could further enhance the robustness and diversity of selected subsets.
- **Fairness and Bias Mitigation:** Incorporating fairness constraints and bias mitigation strategies into the subset selection process to ensure equitable model performance across different groups.
- **Extension to Multimodal Learning:** Adapting DELIFT for multimodal data (e.g., text, images, audio) to expand its applicability beyond natural language processing.
- **Theoretical Analysis:** Developing a deeper theoretical understanding of the utility metric and its properties to further validate and refine the approach.
- **Enhancing Scalability:** Exploring methods to scale DELIFT effectively for larger datasets and more complex models without compromising efficiency.

Our ongoing efforts aim to ensure that DELIFT contributes to responsible and equitable AI development while maximizing efficiency.

G CODE AND DATA AVAILABILITY

To facilitate reproducibility and further research, we will make the DELIFT implementation and the datasets used in our experiments publicly available upon publication. Interested researchers can access these resources through the following repository: https://anonymous.4open.science/r/optimizing-data-selection-0CD0.

H HYPERPARAMETER SETTINGS

Consistent hyperparameter settings were maintained across all experiments to ensure reproducibility:

- **Submodular Function:** Utilized Facility Location (FL), Facility Location Mutual Information (FLMI), or Facility Location Conditional Gain (FLCG) based on the use case.
- **Utility Metric Scaling Factor:** Set $\eta = 1$ for FLMI and $\nu = 1$ for FLCG.
- **Budget (% of Data):** Fixed at 30% for all subset selection experiments.
- **Optimization Algorithm:** Employed greedy maximization with a stopping criterion based on the budget.
- **Distance Metric:** Used length-normalized L2 norm.
- **Teacher Forcing Technique:** Applied during utility metric computation to ensure reliable prediction accuracy measurement.